RECEIVE YOUR AUDIT TRAIL IN

Splunk

Receive the REVIEWIT audit trail in Splunk Enterprise or Splunk Cloud Platform through the HTTP Event Collector: a token you create and entrust, no forwarder, no app to install.

VERSION
1.0 · October 2026

DURATION
Thirty minutes, in three steps

DISTRIBUTION
Confidential · REVIEWIT customers

WHAT YOU NEED

01 **A Splunk administrator** able to enable the HTTP Event Collector and create a token (Settings › Data inputs), and the index where the trail will land.

02 **A subscription administrator** of your REVIEWIT subscription. Creating, activating and replacing a credential ask for a strong authentication of less than ten minutes: the console sends you to your identity provider and brings you back.

03 **The order of this guide**: first the target on your side, then the destination in REVIEWIT, then the test. Nothing is delivered before a real write under your credentials has succeeded.

THE PATH, IN THREE STEPS

STEP 1

Prepare the target

Create the receiving endpoint and the credential REVIEWIT will use — minimal rights, nothing else.

STEP 2

Declare the destination

Four screens in the Subscription Manager: target, residence, content, review.

STEP 3

Test, then activate

A probe write under your credentials; a refusal says why. Delivery starts within the minute.

IN THIS GUIDE

1 How BEACON delivers your audit trail

2 What you receive

3 Prepare the target at Splunk

4 Declare the destination in REVIEWIT

5 Test and activate

6 Verify offline with beacon-verify

7 Operate

8 Troubleshooting and checklist

01 How BEACON delivers your audit trail

BEACON is the delivery channel of the REVIEWIT audit trail. It does not produce the evidence and does not own the events: **the audit trail proves, REVIEWIT SEAL seals, BEACON delivers.** Every audit event becomes a link — a strictly increasing sequence, the SHA-256 of the record, the hash of the previous link, and the original record verbatim — and the head of every chain is signed by the sealing authority. What arrives at your target is a chain you can recompute and a seal you can check, with no call back to REVIEWIT.

YOUR REGION

Audit trail

Events chained per domain and retention class, sealed every hour by REVIEWIT SEAL.

→

YOUR REGION

BEACON

A pass every minute: batches of up to 500 events or 5 MB, a manifest per batch, a heartbeat every 15 minutes.

→

YOUR SIDE

Splunk

The HTTP Event Collector of your Splunk deployment, under a token you entrust to REVIEWIT.

GUIDING PRINCIPLE

Nothing is lost and nothing is pretended. Delivery advances chain by chain, only on a strict acknowledgement; a failure resumes exactly where it stopped; after twelve failed attempts the destination is paused and an alert is raised. A retention purge is delivered as a signed cut, a filter leaves stubs that keep the chain verifiable, and silence is covered by the heartbeat: never an unsigned gap.

SUPPORTED THROUGH

The **Splunk** card of the destination wizard (HTTP Event Collector profile). Splunk Enterprise and Splunk Cloud Platform. Authentication is your HEC token, entrusted once, sealed under REVIEWIT SEAL custody, opened in memory only to deliver. Acknowledgement is strict: a 2xx answer **and** `{"code":0}` in the body; a 200 that says anything else is not an acknowledgement.

02 What you receive

Three kinds of record travel to your target, all as UTF-8 JSON whose property order is the contract. A **link** is one audit event: `metadata.uid` (the stable event identifier), `reviewit.chain` (key, genesis), `reviewit.integrity` (`sequence`, `previous_hash`, `hash`, `raw_data`). A **manifest** closes each batch: chain, range, entry and exit hashes, counts, `batch_id`, and the reference of the latest seal of the chain. A **heartbeat** carries, every 15 minutes, the head and acknowledged positions of every chain.

RECORD	WHERE IT LANDS	WHAT IT IS FOR
`reviewit:audit`	One HEC event per link; `time` is the occurrence time of the audit event	The audit event, with its proof fields and `raw_data` verbatim
`reviewit:audit:manifest`	One HEC event per batch, bare manifest; indexed at delivery time	Range, hashes, counts, `batch_id`, seal reference
`reviewit:audit:heartbeat`	One HEC event every 15 minutes	State of every chain — alert on its absence

AT-LEAST-ONCE, DEDUPLICATED BY YOU

After a network doubt, the same batch may be presented again with the same `batch_id` and the same event identifiers. Dedupe on `metadata.uid`, and order across chains by occurrence time. A record that never arrives is a gap your SIEM can see: every link is numbered, and the heartbeat says where the head stands.

THE PULL, SAME CONTRACT

Your SIEM can also read the trail itself through the REVIEWIT public API, with the scope `audit.read`: `GET /api/v1/audit/chains` lists the chains of your subscription with their start cursor, `GET /api/v1/audit/stream?cursor=…&limit=500&manifest=true` returns the same links and the same manifests as the push. Push and pull verify with the same tool.

03 Prepare the target at Splunk

A Splunk administrator enables the HTTP Event Collector and creates **a token dedicated to REVIEWIT**, bound to the index of your choice. REVIEWIT sets the sourcetype of every event itself; no sourcetype or index extraction needs to be configured in advance.

01 **Settings › Data inputs › HTTP Event Collector › Global Settings**

All Tokens: Enabled. **Enable SSL**: on (BEACON never sends in clear). HTTP port number: **8088** on Splunk Enterprise (the port REVIEWIT allows besides 443). Splunk Cloud Platform listens on `https://http-inputs-<stack>.splunkcloud.com` on 443 — nothing to enable.

02 **New Token**

Name: **reviewit-beacon**. Source name override and description: optional. Leave **Enable indexer acknowledgement** unchecked: REVIEWIT reads the standard `{"text":"Success","code":0}` acknowledgement.

03 **Input Settings**

Source type: **Automatic** — REVIEWIT names its sourcetypes (`reviewit:audit`, `reviewit:audit:manifest`, `reviewit:audit:heartbeat`). Default index: the index that will hold the trail (for example `reviewit_audit`, created beforehand with the retention you want). Allowed indexes: that index only.

04 **Review › Submit, then copy the token value**

The token is a GUID. It is entered once, in the REVIEWIT wizard (section 4), and never shown again on either side. Do not paste it anywhere else.

05 **Self-hosted with Splunk's default certificate: note the fingerprint**

A collector that presents `SplunkServerDefaultCert` (issued by `SplunkCommonCA`) is not publicly trusted. REVIEWIT never disables TLS verification; it lets the administrator **pin** the exact certificate instead (section 7). Have the SHA-256 fingerprint of the collector's certificate at hand, obtained out of band from whoever operates it. A publicly trusted certificate needs nothing.

WHAT YOU WILL ENTER IN THE WIZARD

WIZARD FIELD	VALUE
Transport	**Splunk** (HTTP Event Collector)
Name	How the destination will read in your list — "Splunk — SOC production".
Target URL	The collector's **origin**: `https://<host>:8088` (Enterprise) or `https://http-inputs-<stack>.splunkcloud.com` (Cloud). REVIEWIT appends `/services/collector/event` itself; a full `…/services/collector/event` URL is used as is. No query string, no credentials in the URL.
HEC token	The token value from step 04.
Target geography	Where your Splunk deployment keeps the data (Splunk Cloud: the stack's region). Declarative: an HTTP endpoint cannot be observed.

04 Declare the destination in REVIEWIT

Subscription Manager › **Settings** › **Integrations & AI** › **BEACON** › **New destination**. The wizard has four screens; nothing is created before the last one, and the target is tested against the egress guard before the destination exists.

01 Target

Name the destination, choose the **Splunk** card, enter the target URL and the credential from section 3. The credential is opened in memory for this action only, then sealed under REVIEWIT SEAL custody: the screen never shows it again.

02 Residence

Declare the **geography where the index lives**. For an HTTP endpoint it cannot be observed, so it is declarative. If it differs from the geography of your REVIEWIT subscription, a written justification is required and recorded in the audit trail: the trail leaves its geography, and that is your decision.

03 Content

Start: from the beginning of each chain (full history) or from activation (only what follows). **Events sent in full**: leave empty for the full flow. A filter replaces every excluded event with a stub — sequence and hashes, no business data — and the sealed proof stops at the first stub; the screen says so before you confirm.

04 Review

Check the summary and click **Create**. The egress guard judges the target first: HTTPS only, no credentials in the URL, port 443 or 8088, a publicly routable host. A refused target names the failing check; nothing is created.

THE EGRESS GUARD, BEFORE EVERY SEND

Every target passes the same seven checks at creation, at every test and before every delivery: valid URL · HTTPS · no credentials in the URL · port 443 or 8088 · no internal REVIEWIT zone · host resolves · host is not a private or reserved address. Redirects are never followed, and TLS verification is never disabled.

05 Test and activate

A destination is activated only after a real write under your credentials has succeeded. Open the destination panel and click **Test the target**: REVIEWIT writes one probe record (a test event under your token). "The target passes the egress guard" means the host, the credential and its rights are proven. A refusal says its cause — never the secret.

01 Test the target

REVIEWIT posts one event to the collector under your token and expects `{"code":0}`. A refusal names its cause: **HEC HTTP 403** — token invalid or disabled; **HEC HTTP 400 without acknowledgement** — index not allowed for the token, or a malformed collector URL; **HEC HTTP 404** — the collector path does not exist at this address (check the origin); a TLS refusal — the certificate is not publicly trusted (pin it, section 7).

02 Activate

From the panel, **Activate** (or tick "Activate as soon as it is created" in the wizard). Delivery begins at the next pass, within the minute; the first heartbeat follows.

03 Read the panel

Per chain: the acknowledged checkpoint, the age of the oldest undelivered event, the margin before the next retention cut, the latest seal and what lies beyond it. The batch history lists every manifest; the channel journal lists every administrative gesture with its audit code.

06 Verify offline with beacon-verify

`beacon-verify` is a command-line tool distributed with its public specification. It reads what your target received and the keyring REVIEWIT publishes at `/.well-known/seal-keyring.json`, anchored on the root fingerprint printed in your contract's security annex, and renders a verdict per chain and per range — with no network and no call to REVIEWIT.

THE COMMAND

```
beacon-verify <export> --keys seal-keyring.json --root <root fingerprint> --namespace <y
our sealing namespace>
```

Your **sealing namespace** is shown in the destination panel (`t-` followed by 26 characters). Pass it with `--namespace`: a seal from another namespace is, for you, a break. Without `--root` and `--namespace`, ranges are reported as sealed *with reserve*.

01 **Search over an indexing window, never over `_time`**

`index=reviewit_audit sourcetype="reviewit:audit*" earliest=0 _index_earliest=<epoch of your window start>`. Full links carry the occurrence time, while stubs, cuts and manifests carry the indexing time: a bound on `_time` cuts batches apart and makes gaps appear that do not exist.

02 **Export as Raw Events**

Search results › Export › Format **Raw Events**, no row limit. Do not use the table view: a value that is both raw and wrapped is refused by the verifier (exit code 2).

03 **Run the verifier**

`beacon-verify reviewit_beacon_raw_events.txt --keys seal-keyring.json --root <root> --namespace <t-…>`. On the REVIEWIT proof of 18 September 2026 — 1 727 records exported from Splunk — the chain sealed since activation rendered SEALED (exit 0); chains sealed before the namespace was bound rendered NOT VERIFIABLE, as expected, and became SEALED at their next seal.

VERDICT	MEANING
`SEALED`	Chained up to a head whose signature is valid, keyring anchored, namespace pinned.
`SEALED, WITH RESERVE`	Same, but the root or the namespace were not pinned on the command line.
`CHAIN ONLY`	Intact, beyond the last seal or behind a stub — the next seal will cover it.
`NOT VERIFIABLE`	A seal without namespace, an unknown algorithm or a key absent from the keyring — never called false.
`BROKEN`	A wrong hash, a gap, a divergent duplicate, an invalid signature or a foreign namespace — with the cause and the rank.

Exit codes: **0** nothing broken and everything before the last seal is sealed · **1** a break or a foreign namespace · **2** unusable input (a duplicated JSON property, a value both raw and wrapped, a record truncated at indexing) · **3** nothing broken, proof incomplete.

07 Operate

Every gesture on a destination is made by a subscription administrator, traced in the audit trail under their name, and the sensitive ones ask for a strong authentication of less than ten minutes. The channel that exports the trail is itself under the trail.

Rotate the HEC token	Create the new token in Splunk, then panel › **Replace the HEC token**. The replacement is dry: the previous token stops being used at once. Disable the old token in Splunk afterwards.
Pin the collector's certificate	Panel › Transport › "Certificate presented" shows the subject, issuer, expiry and SHA-256 fingerprint the egress guard read, and what public verification reproaches. Compare the fingerprint with the one the collector's operator gave you, then **Trust this TLS certificate** (strong authentication, traced). Deliveries then accept publicly trusted certificates or exactly this one — nothing else. A renewed collector certificate must be pinned again; until then the destination pauses for failures. The pin goes away with the target when the URL changes.
Pause for failures	A persistent refusal — a revoked credential, a removed right, a target gone — pauses the destination after twelve attempts and raises an alert in the Alerts Center. Nothing is lost: resumption restarts at the acknowledged checkpoint of every chain, automatically when the hourly probe sees the target again, or with **Resume**.
Change the target	**Edit** › new URL, same form. The token is kept; the pinned certificate, if any, is dropped and must be pinned again.
Retire	**Disable**, then remove or archive. A destination that delivered is archived, never deleted: its manifests are delivery proofs. Revoke the credential on your side afterwards: it serves no purpose any more.

08 Troubleshooting

WHAT YOU SEE	CAUSE	WHAT TO DO
Every batch refused with HEC HTTP 403	The token was disabled or deleted, or "All Tokens" is off	Re-enable or recreate the token; replace it in the panel; **Resume**.
HEC HTTP 400 without acknowledgement	The token's allowed indexes do not include its default index, or the index does not exist	Fix the token's index settings in Splunk; **Test the target**.
The egress guard refuses the target: "Only ports 443 and 8088 are allowed"	The collector listens on another port	Move the collector to 8088 (Global Settings) or front it with a reverse proxy on 443.
The egress guard refuses: "The host resolves to a private or reserved address"	The collector is only reachable on your private network	Expose it on a public address, or receive the trail on premises through an S3-compatible store (see the Amazon S3 guide).
A TLS error on every send	Self-signed or private CA certificate on the collector	Pin the certificate from the panel (section 7). Never ask REVIEWIT to disable verification: it cannot.
beacon-verify reports BROKEN on an export that looks complete	The export was bounded by `_time`, or made from the table view	Export by indexing window, as Raw Events; run again. If the break holds on a clean export, contact REVIEWIT support with the chain key and the rank.
Destination paused, "after failures"	Twelve consecutive refusals (token, index, TLS, network)	Read the cause in the panel, fix it, **Test the target**, then **Resume**. Nothing was lost.

Checklist before you start

7 ITEMS

- [ ] HEC enabled with SSL, on port 8088 (Enterprise) or the Cloud `http-inputs` host
- [ ] An index created with the retention you want
- [ ] A token dedicated to REVIEWIT, bound to that index, indexer acknowledgement off
- [ ] The collector reachable on a public address, publicly trusted certificate or its SHA-256 fingerprint at hand
- [ ] The geography of the index known, and its justification if it differs from your subscription's
- [ ] A subscription administrator with their second factor ready
- [ ] The root fingerprint from your contract annex, for beacon-verify

This guide is provided to REVIEWIT customers who connect their SIEM or storage to BEACON. Menu paths are those of the vendors' consoles at publication date; they may change. In case of discrepancy, REVIEWIT support will assist you.